Some New Iterative Methods Based on Composite Trapezoidal Rule for Solving Nonlinear Equations

Oghereyivwe Oghovese$^1$, Emunefe O. John$^2$

$^1$(Department of Mathematics and Statistics, Delta State Polytechnic, Ozoro, Nigeria)
$^2$(Department of General Studies, Petroleum Training Institute, Effurun, Delta State, Nigeria)

ABSTRACT: In this paper, new two steps family of iterative methods of order two and three constructed based on composite trapezoidal rule and fundamental theorem of calculus, for solving nonlinear equations. Several numerical examples are given to illustrate the efficiency and performance of the iterative methods; the methods are also compared with well known existing iterative method.

KEYWORDS: Nonlinear equations, computational order of convergence, Newton’s method

I. INTRODUCTION

Solving nonlinear equations is one of the most predominant problems in numerical analysis. A classical and very popular method for solving nonlinear equations is the Newton’s method. Some historical points on this method can be found in [1]. Recently, some methods have been proposed and analyzed for solving nonlinear equations [2-13]. Some of these methods have been suggested either by using quadrature formulas, homotopy, decomposition or Taylor’s series [2-13]. Motivated by these techniques applied by various authors [2-13] and references therein, in constructing numerous iterative methods for solving nonlinear equations, we suggest a two steps family of iterative method based on composite trapezoidal rule and fundamental theorem of calculus for solving nonlinear equations. We also considered the convergence analysis of these methods. Several examples of functions, some of which are same as in [2-13] were used to illustrate the performance of the methods and comparison with other existing methods.

II. PRELIMINARIES

We use the following definitions:

Definition 1. (See Dennis and Schnable [2]) Let $\alpha \in \mathbb{R}$, $x_n \in \mathbb{R}$, $n = 0, 1, 2, \ldots$ Then, the sequence $\{x_n\}$ is said to converge to $\alpha$ if
\[
\lim_{n \to \infty} |x_n - \alpha| = 0
\]  (1)
If, in addition, there exists a constant $c \geq 0$, an integer $x_0 \geq 0$, and $p \geq 0$ such that for all $n \geq x_0$,
\[
|x_{n+1} - \alpha| \leq c|x_n - \alpha|^p
\]  (2)
then $\{x_n\}$ is said to converge to $\alpha$ with $q$-order at least $p$. If $p = 2$, the convergence is said to be of order 2.

Definition 2 (See Grau-Sanchez et al. [14]) The computational local order of convergence, $\bar{p}_n$, (CLOC) of a sequence $\{x_n\}_{n \geq 0}$ is defined by
\[
\bar{p}_n = \frac{\log|e_n|}{\log|e_{n-1}|},
\]  (3)
where $x_{n-1}$ and $x_n$ are two consecutive iterations near the roots $\alpha$ and $e_n = x_n - \alpha$.

Notation 1: (See [6]) The notation $e_n = x_n - \alpha$ is the error in the nth iteration. The equation
\[
e_{n+1} = ce_n^p + O(e_n^{p+1})
\]  (4)
is called the error equation. By substituting $e_n = x_n - \alpha$ for all $n$ in any iterative method and simplifying, we obtain the error equation for that method. The value of $p$ obtained is called the order of this method.
III. DEVELOPMENT OF THE METHODS

Consider a nonlinear equation
\[ f(x) = 0 \] (5)

By the Fundamental Theorem of Calculus, if \( f(x) \) is continuous at every point of \([a,b]\) and \( F \) is any antiderivatives of \( f(x) \) on \([a,b]\), then
\[ \int_a^b f(x) \, dx = F(b) - F(a) \] (6)

Differentiating both side of (6) with respect to \( x \), we have;
\[ f(x) = f(b) - f(a) \] (7)

where \( f(b) \) and \( f(a) \) are derivatives of \( F(b) \) and \( F(a) \) respectively.

Recall the Composite Trapezoidal rule given by;
\[ \int_a^b f(x) \, dx = \frac{b-a}{2n} \left[ f(a) + 2 \sum_{i=1}^{n-1} f_i + f(b) \right] \] (8)

If \( n = 2 \) in (8) we have;
\[ \int_a^b f(x) \, dx = \frac{b-a}{4} \left[ f(a) + 2f \left( \frac{a+b}{2} \right) + f(b) \right] \] (9)

Differentiating (9) with respect to \( x \), we have;
\[ f(x) = \frac{b-a}{4} \left[ f'(a) + 2f \left( \frac{a+b}{2} \right) + f'(b) \right] \] (10)

Equating (7) and (10) we have;
\[ f(b) - f(a) = \frac{b-a}{4} \left[ f'(a) + 2f \left( \frac{a+b}{2} \right) + f'(b) \right] \] (11)

From (5), we have;
\[ x = a - 4 \frac{f(a)}{f'(a)} - (x-a) \frac{f'(x)}{f'(a)} - 2(x-a) \frac{f'(a)}{f'(a)} \] (12)

Using (12), one can suggest the following iterative method for solving the nonlinear equation (5).

**Algorithm 1:** Given an initial approximation \( x_0 \) (close to \( \alpha \) the root of (5)), we find the approximate solution \( x_{n+1} \) by the implicit iterative method:
\[ x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - (x-a) \frac{f'(x_n)}{f'(a)} - 2(x-a) \frac{f'(a)}{f'(a)}, \quad n = 0, 1, 2, \ldots \] (13)

The implicit iterative method in (13) is a predictor-corrector scheme, with Newton’s method as the predictor, and Algorithm 1 as the corrector. The first consequence of (13) is the suggested two-step iterative method for solving (5) stated as follows:

**Algorithm 2:** Given an initial approximation \( x_0 \) (close to \( \alpha \) the root of (5)), we find the approximate solution \( x_{n+1} \) by the iterative schemes:
\[ y_n = x_n - \frac{f(x_n)}{f'(x_n)} \] (14a)
\[ x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - (y_n - x_n) \frac{f'(y_n)}{f'(x_n)} - 2(y_n - x_n) \frac{f'(x_n)}{f'(x_n)}, \quad n = 0, 1, 2, \ldots \] (14b)

From (14a), we have that;
\[ y_n = x_n - \frac{f(x_n)}{f'(x_n)} \] (15)

Using (15) in (14b) we suggest another new iterative scheme as follows:

**Algorithm 3:** Given an initial approximation \( x_0 \) (close to \( \alpha \) the root of (5)), we find the approximate solution \( x_{n+1} \) by the iterative schemes:
Some New Iterative Methods Based on Composite...

\[ x_{n+1} = x_n - 4 \frac{f(x_n)}{f'(x_n)} + \frac{f(x_n)}{f'(x_n)} \left( \frac{f'(y_n)}{f'(x_n)} - 2 \right) \left( \frac{f(x_n) + y_n}{f(x_n)} \right), \quad n = 0, 1, 2, \ldots \] (16)

From (5) and (11) we can have the fixed point formulation given by

\[ x = a - \frac{4f(a)}{f'(a) + 2f' \left( \frac{a + x}{2} \right) + f(x)} \] (17)

The formulation (17) enable us to suggest the following iterative method for solving nonlinear equations.

**Algorithm 4:**

Given an initial approximation \( x_0 \) close to \( a \) the root of (5), we find the approximate solution \( x_{n+1} \) by the iterative schemes:

\[ y_n = x_n - \frac{f(x_n)}{4f(x_n)}, \quad n = 0, 1, 2, \ldots \]

\[ x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n) + 2f' \left( \frac{x_n + y_n}{2} \right) + f(y_n)}, \quad n = 0, 1, 2, \ldots \] (18)

In the next section, we present the convergence analysis of Algorithm 2 and 4. Similar procedures can be applied to analyze the convergence of Algorithm 3.

IV. CONVERGENCE ANALYSIS OF THE METHODS

Theorem 1: Let \( \alpha \in I \) be a simple zero of sufficiently differentiable function \( f:I \subseteq \mathbb{R} \rightarrow \mathbb{R} \) for an open interval \( I \). If \( x_0 \) is sufficiently close to \( \alpha \), then the iterative method defined by (14) is of order two and it satisfies the following error equation:

\[ e_{n+1} = \alpha - 3c_2 e_n^2 + \left( c_3 + 6c_2^2 - \frac{3}{2} c_2 \right) e_n^3 + O(e_n^3) \] (19)

where

\[ c_2 = \frac{f''(\alpha)}{2f'(\alpha)} \] (20)

**Proof**

Let \( \alpha \) be a simple zero of \( f \), and \( e_n = x_n - \alpha \). Using Taylor expansion around \( x = \alpha \) and taking into account \( f(\alpha) = 0 \), we get

\[ f(x_n) = f'(\alpha) [e_n + c_2 e_n^2 + c_3 e_n^3 + c_4 e_n^4 + \ldots], \] (21)

\[ f'(x_n) = f'(\alpha) [1 + 2c_2 e_n + 3c_3 e_n^2 + 4c_4 e_n^3 + 5c_5 e_n^4 + \ldots] \] (22)

where \( c_k = \frac{f^{(k)}(\alpha)}{k! f'(\alpha)} \), \( k = 2, 3, 4, \ldots \) (23)

Using (21) and (22), we have;

\[ \frac{f(x_n)}{f'(x_n)} = \left[ e_n - c_2 e_n^2 + 2(c_2^2 - c_3) e_n^3 + (7c_2c_3 - 4c_2^3 - 3c_4) e_n^4 + \ldots \right] \] (24)

But

\[ y_n = x_n - \frac{f(x_n)}{f'(x_n)} \] (25)

\[ = \left[ \alpha + c_2 e_n^2 + 2(c_2^2 - c_3) e_n^3 - (7c_2c_3 - 4c_2^3 - 3c_4) e_n^4 + \ldots \right] \] (26)

Hence,

\[ y_n - x_n = -e_n + c_2 e_n^2 + 2(c_2^2 - c_3) e_n^3 - (7c_2c_3 - 4c_2^3 - 3c_4) e_n^4 + \ldots \] (27)

From (26), we have;

\[ f'(y_n) = f'(\alpha) [1 + 2c_2 e_n^2 + 4(c_2 c_3 - c_2^3) e_n^3 + (-11c_2^2 c_3 + 6c_2 c_4 + 8c_2^4) e_n^4 + \ldots] \] (28)

Combining (22) and (28), we have;

\[ \frac{f'(y_n)}{f'(x_n)} = -2c_2 e_n + (-3c_3 + 6c_2^2) e_n^3 + (-16c_2^3 - 4c_4 + 16c_2 c_3) e_n^4 + \ldots \] (29)

From (27) and (29) we have;

\[ (y_n - x_n) \frac{f'(y_n)}{f'(x_n)} = -e_n + 3c_2 e_n^2 + (5c_3 - 10c_2^2) e_n^3 + (-30c_2 c_3 + 30c_2^3 + 7c_4) e_n^4 + \ldots \] (30)

www.ijmsi.org 3 | P a g e
From the relation:
\[
\frac{x_n + y_n}{2} = x_n - f(x_n) \quad \frac{2}{2f(x_n)}
\]
\[
= \alpha + \frac{1}{2}c_3 + \frac{3}{4}c_2e_n^2 - (c_2 - c_3)e_n^2 - \frac{1}{2}(7c_2c_3 - 4c_2^3 - 3c_3)e_n^4 + \cdots \tag{31}
\]
we have:
\[
f\left(\frac{x_n + y_n}{2}\right) = f\left(x_n - f\left(\frac{x_n}{2}\right)\right)
\]
\[
= f(\alpha) \left(1 + c_2e_n + c_3 + c_2e_n^2 \left(-\frac{1}{2}c_2 + \frac{9}{8}c_3\right) e_n^3 + \left(\frac{5}{4}c_2^3 - \frac{17}{8}c_2c_3 + \frac{25}{16}c_4\right) e_n^4 + \cdots \right)
\]
\[
+ \frac{3c_2^3 + \frac{57}{8}c_3c_2^3 - \frac{9}{4}c_4c_2 - \frac{13}{4}c_2c_4 + \frac{65}{32}c_5}{c_2c_3} e_n^5 + \cdots \tag{32}
\]
\[
f'[\left(\frac{x_n + y_n}{2}\right)] = f(\alpha) \left(1 + c_2e_n + c_3 + c_2e_n^2 \left(-\frac{7}{2}c_2c_3 + \frac{1}{2}c_4\right) e_n^3 + \cdots + \frac{9}{2}c_2c_4 + c_2^3 + \frac{37}{4}c_2c_3 + 3c_2^3 + \frac{5}{16}c_3\right) e_n^4 + \cdots \tag{33}
\]
Using (22) and (33) we have:
\[
f'[\left(\frac{x_n + y_n}{2}\right)] f(\alpha) = 1 - c_2e_n \left(3c_2 - \frac{3}{4}c_3 - 3c_3\right) e_n^2 + \cdots \tag{34}
\]
And (27) with (34) gives:
\[
(y_n - x_n) \frac{f'[\left(\frac{x_n + y_n}{2}\right)]}{f(\alpha)} = -e_n + 2c_2e_n^2 + \left(\frac{3}{4}c_2 - c_3 - 2c_3\right) e_n^3 + \cdots \tag{35}
\]
Using (29), (30) and (35) in
\[
x_{n+1} = x_n - 4 \frac{f(x_n)}{f'(x_n)} - (y_n - x_n) \frac{f'(y_n)}{f'(x_n)} - 2(y_n - x_n) \frac{f'[\left(\frac{x_n + y_n}{2}\right)]}{f(\alpha)}
\]
\[
= \alpha - 3c_2e_n^2 + \left(c_3 + \frac{6c_2^3}{2} - \frac{3}{2}c_2\right) e_n^3 + O(e_n^4) \tag{36}
\]
Thus, we observe that the Algorithm 2 is second order convergent.

**Theorem 2:** Let \( \alpha \in I \) be a simple zero of sufficiently differentiable function \( f: I \subseteq \mathbb{R} \to \mathbb{R} \) for an open interval \( I \). If \( x_0 \) is sufficiently close to \( \alpha \), then the iterative method defined by (18) is of order three and it satisfies the following error equation:
\[
e_n^{n+1} = \alpha + \left(\frac{1}{8}c_3 + c_2^2\right) e_n^3 + O(e_n^4) \tag{37}
\]
where \( c_3 = \frac{f'(\alpha)}{3f(\alpha)} \tag{38} \)

**Proof** Using (22), (33) and (28) we have:
\[
f(x_n) + 2f'\left(\frac{x_n + y_n}{2}\right) + f'(y_n) = f'(\alpha) \left[4 + 4c_2e_n + \left(\frac{9}{2}c_2 + 4c_3^2\right) e_n^2 + \left(5c_4 + 11c_2c_3 - 8c_2^3\right) e_n^3 + (-3c_2^3) e_n^4 + \cdots \right] \tag{39}
\]
and from (21) we have:
\[
4f'(x_n) = f'(\alpha) \left[4e_n + 4c_2e_n^2 + 4c_3e_n^3 + 4c_4e_n^4 + \cdots \right] \tag{40}
\]
Combining (39) and (40) in (18) gives;

---

[www.ijmsi.org](http://www.ijmsi.org) 4 | Page
\[ e_{n+1} = x_n - \frac{4f(x_n)}{f'(x_n) + 2f\left(\frac{x_n + y_n}{2}\right) + f'(y_n)} = \alpha + \left(\frac{1}{6}c_3 + \frac{c_2}{2}\right)e_n^3 + O(e_n^4) \]  

(41)

This means the method defined by (18) is of third-order. That completes the proof.

V. NUMERICAL EXAMPLES

In this section, we present some examples to illustrate the efficiency of our developed methods which are given by the Algorithm 1 – 4. We compare the performance of Algorithm 2 (AL2) and Algorithm 4 (AL4) with that of Newton Method (NM). All computations are carried out with double arithmetic precision. Displayed in Table 1 are the number of iterations (NT) required to achieve the desired approximate root \(x_n\) and respective Computational Local Order of Convergence (CLOC), \(\rho_n\). The following stopping criteria were used.

\[ i. \quad |x_{n+1} - x_n| < \varepsilon \quad \text{and} \quad ii. \quad f(x_{n+1}) < \varepsilon \]  

(37)

where \(\varepsilon = 10^{-15}\).

We used the following functions, some of which are same as in [2-4,6-12,14]

\[
\begin{align*}
&f_1(x) = (x - 1)^3 - 1 \\
&f_2(x) = \cos(x) - x \\
&f_3(x) = x^3 - 10 \\
&f_4(x) = x^2 - e^x - 3x + 2 \\
&f_5(x) = (x + 2)e^x - 1 \\
&f_6(x) = x^3 + 4x^2 - 10 \\
&f_7(x) = \ln x + \sqrt{x} - 5 \\
&f_8(x) = e^x \sin x + \ln(\sqrt{x^2 + 1})
\end{align*}
\]  

(38)

Table 1: Comparison between methods depending on the number of iterations (IT) and Computational Local Order of Convergence.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>( x_0 )</th>
<th>( f(x) )</th>
<th>( \text{Number of iterations (NT)} )</th>
<th>( \text{Computational Local Order of Convergence (CLOC)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>( \text{NM} )</td>
<td>( \text{AL2} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The computational results presented in Table 1 shows that the suggested methods are comparable with Newton Method. This means that; the new methods (Algorithm 4 in particular) can be considered as a significant improvement of Newton Method, hence; they can serve as an alternative to other second and third order convergent respectively, methods of solving nonlinear equations.

VI. CONCLUSION

We derived a two step family of iterative methods based on composite trapezoidal rule and fundamental theorem of calculus, for solving nonlinear equations. Convergence proof is presented in detail for algorithm 2 and 4 and they are of order two and three respectively. Analysis of efficiency showed that these methods can be used as alternative to other existing order two and three iterative methods for zero of nonlinear equations. Finally, we hoped that this study makes a contribution to solve nonlinear equations.
REFERENCES


